(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号
特開2001－313634
（P2001－313634A）
(43)公開日　平成13年11月９日(2001.11.9)

(51)Int.Cl.[7]	識別記号	ＦＩ		ﾃｰﾏｺｰﾄﾞ(参考)
Ｈ０４Ｌ　9/08		Ｇ０６Ｆ　15/00	３３０Ｅ	
Ｇ０６Ｆ　15/00	３３０	Ｈ０４Ｌ　9/00	６０１Ｃ	
Ｈ０４Ｌ　9/32			６７３Ａ	

審査請求　未請求　請求項の数33　ＯＬ　（全 16 頁）

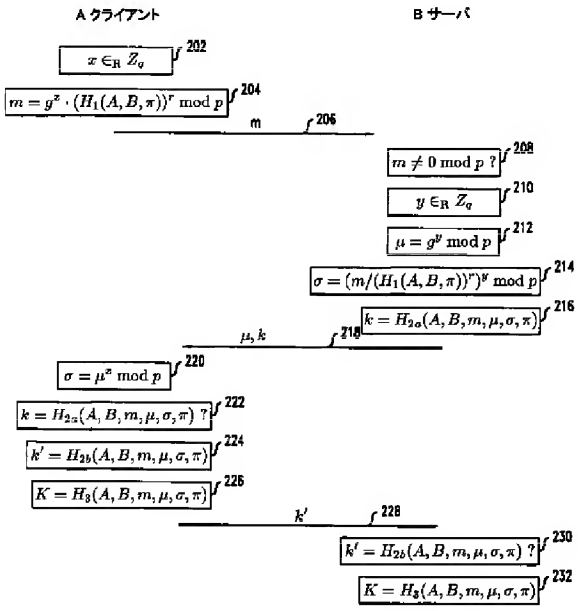
(21)出願番号　特願2001－71600(P2001－71600)
(22)出願日　平成13年３月14日(2001.3.14)
(31)優先権主張番号　６０／１９０３１８
(32)優先日　平成12年３月17日(2000.3.17)
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　０９／６３８３２０
(32)優先日　平成12年８月14日(2000.8.14)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　596077259
ルーセント　テクノロジーズ　インコーポレイテッド
Ｌｕｃｅｎｔ　Ｔｅｃｈｎｏｌｏｇｉｅｓ　Ｉｎｃ.
アメリカ合衆国　07974　ニュージャージー、マレーヒル、マウンテン　アベニュー　600－700
(74)代理人　100081053
弁理士　三俣　弘文

最終頁に続く

(54)【発明の名称】　通信方法

(57)【要約】
【課題】　安全であることが証明可能な、安全な、パスワードのみによる相互ネットワーク認証プロトコルを提供する。
【解決手段】　データネットワークを通じて、パスワードを共有するＡ、Ｂ二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密g^{xy}を生成する。ｇは両当事者に既知の群生成元であり、ｘは一方当事者Ａ（例えばクライアント）に既知の指数であり、ｙは他方当事者Ｂ（例えばサーバ）に既知の指数である。Ａが、g^{x}と、少なくともパスワードの関数H_1とに対する群演算を実行することによりパラメータｍを生成し、ｍをＢに送信する。これにより、Ｂは、ｍと、関数H_1とに対して群逆演算を実行してg^{x}を抽出し、さらに、共有秘密g^{xy}を計算することが可能となる。Ａも、Ｂからg^{y}を受信し、共有秘密g^{xy}を生成することができる。



【特許請求の範囲】

【請求項１】　データネットワークを通じて、パスワードを共有する二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密g^{xy}を生成する通信方法において、

ｇは両当事者に既知の群生成元であり、ｘは一方当事者に既知の指数であり、ｙは他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、

一方当事者が、g^{x}と、少なくとも前記パスワードの関数とに対する群演算を実行することによりパラメータｍを生成し、ｍを他方当事者に送信するステップを有し、これにより、他方当事者は、ｍと、少なくとも前記パスワードの前記関数とに対して群逆演算を実行してg^{x}を抽出し、さらに、前記共有秘密g^{xy}を計算することが可能であることを特徴とする通信方法。

【請求項２】　前記一方当事者はクライアントであり、前記他方当事者はサーバであることを特徴とする請求項１記載の方法。

【請求項３】　前記一方当事者が、前記他方当事者からg^{y}を受信し、前記共有秘密g^{xy}を生成するステップをさらに有することを特徴とする請求項１記載の方法。

【請求項４】　前記一方当事者が、受信値を、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^{y}、前記共有秘密、および前記パスワードのうちの少なくとも１つの関数と比較することによって、前記他方当事者を認証するステップをさらに有することを特徴とする請求項３記載の方法。

【請求項５】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^{y}、前記共有秘密、および前記パスワードのうちの少なくとも１つの関数を、前記他方当事者に送信するステップをさらに有し、これにより、前記他方当事者は、前記一方当事者を認証することが可能であることを特徴とする請求項３記載の方法。

【請求項６】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^{y}、前記共有秘密、および前記パスワードのうちの少なくとも１つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項３記載の方法。

【請求項７】　g^{y}と、少なくとも前記パスワードの関数とに対して群演算を実行することにより前記他方当事者が計算したパラメータμを、前記一方当事者が受信するステップと、

前記一方当事者が、μと、少なくとも前記パスワードの前記関数とに対して群逆演算を実行してg^{y}を抽出するステップと、

前記一方当事者が、前記共有秘密g^{xy}を計算するステップとをさらに有することを特徴とする請求項１記載の方法。

【請求項８】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、μ、前記共有秘密、および前記パスワードのうちの少なくとも１つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項７記載の方法。

【請求項９】　データネットワークを通じて、二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密g^{xy}を生成する通信方法において、

ｇは両当事者に既知の群生成元であり、ｘは一方当事者に既知の指数であり、ｙは他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、

一方当事者が、g^{x}と、少なくともパスワードベリファイアの関数とに対する群演算を実行することによりパラメータｍを生成し、ｍを他方当事者に送信するステップを有し、これにより、他方当事者は、ｍと、少なくとも前記パスワードベリファイアの前記関数とに対して群逆演算を実行してg^{x}を抽出し、さらに、前記共有秘密g^{xy}を計算することが可能であることを特徴とする通信方法。

【請求項１０】　前記一方当事者はクライアントであり、前記他方当事者はサーバであることを特徴とする請求項９記載の方法。

【請求項１１】　前記一方当事者が、前記他方当事者からg^{y}を受信し、前記共有秘密g^{xy}を生成するステップをさらに有することを特徴とする請求項９記載の方法。

【請求項１２】　前記一方当事者が、受信値を、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^{y}、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも１つの関数と比較することによって、前記他方当事者を認証するステップをさらに有することを特徴とする請求項１１記載の方法。

【請求項１３】　前記一方当事者に既知のインデックスをＣとして、前記一方当事者が、g^{C}を計算するステップと、

前記一方当事者が、少なくともg^{C}と、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^{y}、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも１つとの関数としてパラメータｅを計算するステップと、

前記パスワードベリファイアの離散対数をｖとして、前記一方当事者が、Ｓ＝Ｃ－ｅｖとしてパラメータＳを計算するステップと、

前記一方当事者が、Ｓおよびg^{C}を前記他方当事者へ送信するステップとをさらに有し、これにより、前記他方当事者は、前記Ｓおよびg^{C}の値に少なくとも部分的に基づいて前記一方当事者を認証することが可能であることを特徴とする請求項１２記載の方法。

【請求項１４】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^{y}、前記共有秘密、g^{C}および前記パスワードベリファイアのうちの

少なくとも１つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項１３記載の方法。

【請求項１５】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^y、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも１つの関数を、前記他方当事者に送信するステップをさらに有し、これにより、前記他方当事者は、前記一方当事者を認証することが可能であることを特徴とする請求項１１記載の方法。

【請求項１６】　前記パスワードベリファイアを公開鍵とし、前記パスワードベリファイアの離散対数を秘密鍵として用いてランダム値が暗号化される自己証明エルガマル暗号に基づいて、前記一方当事者は前記他方当事者を認証することを特徴とする請求項１１記載の方法。

【請求項１７】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^y、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも１つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項１６記載の方法。

【請求項１８】　データネットワークを通じて、パスワードを共有する二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密g^{xy}を生成する通信方法において、

ｇは両当事者に既知の群生成元であり、ｙは一方当事者に既知の指数であり、

ｘは他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、g^xと、少なくとも前記パスワードの関数とに対して群演算を実行することにより前記他方当事者が計算したパラメータｍを、前記一方当事者が受信するステップと、

前記一方当事者が、ｍと、少なくとも前記パスワードの前記関数とに対して群逆演算を実行してg^xを抽出し、さらに、前記共有秘密g^{xy}を計算するステップとを有することを特徴とする通信方法。

【請求項１９】　前記一方当事者は前記パスワードを記憶することを特徴とする請求項１８記載の方法。

【請求項２０】　前記一方当事者は、少なくとも前記パスワードの前記関数を表す値を記憶するが、前記一方当事者は前記パスワードを記憶しないことを特徴とする請求項１８記載の方法。

【請求項２１】　前記一方当事者はサーバであり、前記他方当事者はクライアントであることを特徴とする請求項１８記載の方法。

【請求項２２】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^y、前記共有秘密、および前記パスワードのうちの少なくとも１つの関数を、前記他方当事者に送信するステップをさらに有することを特徴とする請求項１８記載の方法。

【請求項２３】　前記一方当事者が、受信値を、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^y、前記共有秘密、および前記パスワードのうちの少なくとも１つの関数と比較することによって、前記他方当事者を認証するステップをさらに有することを特徴とする請求項１８記載の方法。

【請求項２４】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^y、前記共有秘密、および前記パスワードのうちの少なくとも１つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項１８記載の方法。

【請求項２５】　前記一方当事者が、g^yと、少なくとも前記パスワードの関数とに対する群演算を実行することによりパラメータμを生成し、μを前記他方当事者に送信するステップを有し、これにより、前記他方当事者は、μと、少なくとも前記パスワードの前記関数とに対して群逆演算を実行してg^yを抽出し、さらに、前記共有秘密g^{xy}を計算することが可能であることを特徴とする請求項１８記載の方法。

【請求項２６】　データネットワークを通じて、二当事者間で、特定の群に関するDiffie-Hellman型鍵交換を用いて共有秘密g^{xy}を生成する通信方法において、

ｇは両当事者に既知の群生成元であり、ｙは一方当事者に既知の指数であり、

ｘは他方当事者に既知の指数であり、前記群は群演算および群逆演算を有し、g^xと、少なくともパスワードベリファイアの関数とに対して群演算を実行することにより前記他方当事者が計算したパラメータｍを、前記一方当事者が受信するステップと、

前記一方当事者が、ｍと、少なくとも前記パスワードベリファイアの前記関数とに対して群逆演算を実行してg^xを抽出し、さらに、前記共有秘密g^{xy}を計算するステップとを有することを特徴とする通信方法。

【請求項２７】　前記一方当事者はサーバであり、前記他方当事者はクライアントであることを特徴とする請求項２６記載の方法。

【請求項２８】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^y、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも１つの関数を、前記他方当事者に送信するステップをさらに有することを特徴とする請求項２６記載の方法。

【請求項２９】　Ｃはランダム値であり、

ｖは、前記パスワードベリファイアの離散対数であり、

Ｓ＝Ｃ－ｅｖであり、

ｅは、少なくともg^Cと、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、g^y、前記共有秘密、および前記パスワードベリファイアのうちの少なくとも１つとの関数であるとして、

前記一方当事者が、前記他方当事者から受信したパラメ

ータＳおよびｇ^{c}の値に少なくとも部分的に基づいて、前記他方当事者を認証するステップをさらに有することを特徴とする請求項２６記載の方法。

【請求項３０】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、ｇ^{y}、前記共有秘密、ｇ^{c}および前記パスワードベリファイアのうちの少なくとも１つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項２９記載の方法。

【請求項３１】　前記一方当事者が、前記パスワードベリファイアを公開鍵とし、前記パスワードベリファイアの離散対数を秘密鍵として用いる自己証明エルガマル暗号を用いてランダム値を暗号化するステップと、
暗号化されたランダム値を前記他方当事者へ送信するステップとをさらに有することを特徴とする請求項２６記載の方法。

【請求項３２】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、ｇ^{y}、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも１つの関数として前記他方当事者により計算された受信値に基づいて前記他方当事者を認証するステップをさらに有することを特徴とする請求項２６記載の方法。

【請求項３３】　前記一方当事者が、前記一方当事者の識別子、前記他方当事者の識別子、ｍ、ｇ^{y}、前記共有秘密、前記ランダム値、および前記パスワードベリファイアのうちの少なくとも１つの関数としてセッション鍵を生成するステップをさらに有することを特徴とする請求項２６記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ネットワーク認証および鍵交換に関し、特に、パスワードのみによる安全な相互ネットワーク認証および鍵交換プロトコルに関する。

【０００２】

【従来の技術】ネットワークを通じての認証は、リモートクライアントがネットワークサーバにアクセスすることを可能にするシステムのセキュリティ（安全性）の重要な部分である。認証は一般に、次のうちの１つまたは複数のものを確認することにより実行される。

・ユーザが知っているもの。例えば、パスワード。

・ユーザ自体。すなわち、指紋のような生体計測情報。

・ユーザが持っているもの。すなわち、スマートカードのような何らかの識別トークン。

例えば、現金自動預入支払機（ＡＴＭ）は、これらのうちの２つ、すなわち、ユーザが持っているもの（ＡＴＭカード）と、ユーザが知っているもの（個人識別番号（暗証番号、ＰＩＮ））とを確認する。ＡＴＭ認証は、データネットワークを通じての認証よりもずっと容易である。その理由は、ＡＴＭ自体は信頼されたハードウェアとみなされ、ＡＴＭカードの存在を確認し正しい情報を安全に中央取引サーバに転送することが信用されているからである。

【０００３】認証に加えて、鍵交換は、データネットワークを通じての通信の重要な部分である。クライアントとサーバが認証された後、安全な通信チャネルがそれらの間に設定されなければならない。これは一般に、クライアントとサーバが、認証に続く通信の間に用いるためのセッション鍵と呼ばれるある鍵を交換することによって実行される。

【０００４】データネットワーク、特に、インターネットのような公衆データネットワークを通じての認証は困難である。その理由は、クライアントとサーバの間の通信は、多くの異なるタイプの攻撃を受けやすいからである。例えば、盗聴(eavesdropping)攻撃では、敵は、クライアントとサーバの間の通信を傍受することにより、秘密情報を知る可能性がある。敵は、パスワード情報を知った場合、その情報をサーバに対して再生（リプレイ）して正当なクライアントになりすますことが可能となる。これをリプレイ攻撃という。リプレイ攻撃は、クライアントから送られたパスワードが暗号化されている場合でも有効である。その理由は、敵は、実際のパスワードを知る必要はなく、代わりに、サーバが正当なクライアントから期待しているもの（この場合、暗号化されたパスワード）をサーバに提供すればよいからである。もう１つのタイプの攻撃は、スプーフィング攻撃である。これは、敵がサーバになりすますことにより、クライアントは、正当なサーバと通信していると信じるが、実際のところは敵と通信しているというものである。このような攻撃では、クライアントは、要注意(sensitive)情報を敵に提供してしまう可能性がある。

【０００５】さらに、パスワードに基づく認証プロトコルでは、パスワードが辞書攻撃を受けるくらい弱くなる可能性が存在する。辞書攻撃は、必要なパスワードに関する何らかの既知情報に対して、多数のありそうなパスワード（例えば、英語辞書中の全単語）をテストすることによって実行される、パスワードに対する力ずくの攻撃である。この既知情報は、公に利用可能なこともあるし、上記の技術のうちの１つにより敵が入手したものであることもある。ユーザはしばしば、覚えやすい、したがって推測しやすいパスワードを選択するので、辞書攻撃はしばしば有効である。

【０００６】ネットワーク認証にはさまざまな技術が知られている。これらの公知技術は２つに分類される。第１の分類には、クライアントシステム上に永続的記憶データを要求する技術が含まれる。第２の分類には、クライアントシステム上に永続的記憶データを要求しない技術が含まれる。

【０００７】第１の分類に関して、永続的記憶データに

は、決して漏れてはならない秘密(secret)データ（例えば、認証サーバと共有される秘密鍵）も、改竄を防止しなければならない、秘密ではないが要注意の(sensitive)データ（例えば、認証サーバの公開鍵）も含まれる。いずれのタイプの永続的データでも、データを敵からの攻撃から守るには、追加のセキュリティ条件が必要である。さらに、パスワードおよび永続的記憶データの両方に依拠する認証プロトコルを用いるときには、一方の漏洩が他方の弱点となることがある。例えば、秘密鍵が漏洩すると、パスワードに対する辞書攻撃が可能となることがある。この第１のクラスのプロトコルでのもう１つの問題点は、永続的記憶データは、鍵の生成および配送を必要とするが、これは面倒なことがあり、一般にシステムの柔軟性が低下することである。

【０００８】第２の分類は、クライアントにおける永続的記憶データを要求しないため、パスワードのみによる認証プロトコルと呼ばれる。クライアントは、正当なパスワードを提供することができればよい。潜在的に弱いパスワードを用いて強力なセキュリティおよび認証を提供するということは、矛盾しているように思われる。しかし、安全であるように設計されたパスワードのみによるユーザ認証および鍵交換のプロトコルがいくつか存在する。これらのプロトコルは、D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer Communication Review, ACM SIGCOMM, 26(5):5-20, 1996、に記載されている。これらのパスワードのみによるプロトコルのうち注目すべきものには次のものがある。

・ＥＫＥ(Encrypted Key Exchange) (S. M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks", Proceedings of the IEEE Symposium on Research in Security and Privacy, pp.72-84、に記載)

・Ａ－ＥＫＥ(Augmented-EKE) (S. M. Bellovin and M. Merritt, "AugmentedEncrypted Key Exchange: A Password-Based Protocol Secure Against Dictionary Attacks and Pasword File Compromise", Proceedings of the First Annual Conference on Computer and Communications Security, 1993, pp.244-250、に記載)

・Ｍ－ＥＫＥ(Modified EKE) (M. Steiner, G. Tsudik, and M. Waidner, "Refinement and Extension of Encrypted Key Exchange", ACM Operating SystemReview, 29:22-30, 1995、に記載)

・ＳＰＥＫＥ(Simple Password EKE)およびＤＨ－ＥＫＥ(Diffie-Hellman EKE) (いずれも、D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer Communication Review, ACM SIGCOMM, 26(5):5-20, 1996、に記載)

・ＳＲＰ(Secure Remote Password Protocol) (T. Wu, "The Secure Remote Password Protocol", Proceedings of the 1998 Internet Society Network andDistributed System Security Symposium, pp.97-111, 1998、に記載)

・ＯＫＥ(Open Key Exchange) (Stefan Lucks, "Open Key Exchange: How toDefeat Dictionary Attacks Without Encrypting Public Keys", Security Protocol Workshop, Ecole Normale Superieure, April 7-9, 1997、に記載)

【０００９】

【発明が解決しようとする課題】これらの公知のパスワードのみによる認証プロトコルでの問題点は、これらは安全であることが証明されていないことである。実際、S. Patel, "Number Theoretic Attacks on Secure Password Schemes", Proceedings of the IEEE Symposium on Research in Security and Privacy, pp.236-247, 1997、に記載されているように、ＥＫＥプロトコルは、いくつかの数論的攻撃を受ける可能性がある。ネットワークセキュリティの重要性の観点から、安全であることが証明可能な、パスワードのみによる相互認証プロトコルが必要である。

【００１０】米国特許出願第０９／３５３，４６８号（出願日：１９９９年７月１３日）には、公開鍵暗号方式を利用した安全なパスワードのみによる相互ネットワーク認証プロトコルが記載されている。このプロトコルは、基礎となる公開鍵暗号方式と同程度に安全であることが証明されている。

【００１１】

【課題を解決するための手段】本発明は、安全であることが証明可能な、安全な、パスワードのみによる相互ネットワーク認証プロトコルを提供する。本発明のプロトコルによれば、二当事者が、Diffie-Hellman型の鍵交換を用いて共有秘密を生成する。以下で詳細に説明するように、Diffie-Hellman型の鍵交換によれば、特定のいわゆる群に対する群生成元ｇと、一方当事者に既知の指数ｘと、他方当事者に既知の指数ｙと、共有秘密g^{xy}がある。一方当事者がg^{x}を生成し、他方当事者がg^{y}を生成し、両者はこれらの値を交換して、それぞれが共有秘密g^{xy}を生成することができるようにする。Diffie-Hellmanは鍵交換プロトコルを定義しているが、このプロトコルは認証の観点を有しない。

【００１２】本発明によれば、Diffie-Hellman型の共有秘密を用いるが、二当事者が共有パスワードを用いて互いを認証することができるように修正されたプロトコルが実現される。さらに、重要な点であるが、本発明の発明者は、このプロトコルが安全であることを証明した。本発明によれば、一方当事者がDiffie-Hellman値g^{x}を生成し、これを、いわゆる群演算を用いて少なくともパスワードの関数と結合し、その結果得られる値を他方当事者へ送信する。群演算は、用いられる特定の群に対して定義され、以下で詳細に説明する。今の目的のために

は、あらゆる群は１つの群演算と、対応する群逆演算とを有することを認識すれば十分である。他方当事者は、値を受信すると、受信した値と、少なくともパスワードの関数とに対して群逆演算を実行して、g^xを抽出する。これにより、他方当事者は、自己のｙの知識を用いて、共有秘密g^{xy}を生成することができる。ここに記載するように、群演算および群逆演算をDiffie-Hellman型の鍵交換プロトコルとともに用いることは、従来技術のパスワードのみによる相互ネットワーク認証プロトコルに比べて利点がある。おそらく最も重要な点は、通信チャネルにアクセス可能な敵による攻撃に対して安全であることを証明することができるプロトコルを提供する点である。上記のように、Diffie-Hellman値g^xは、少なくともパスワードの関数と結合される。「少なくとも」という用語を用いるのは、さまざまな実施例において、g^xは、パスワードのみの関数と結合されることもあるが、パスワードが特定の当事者のペアに対して一意的であることを保証するために、パスワードと、プロトコルに対する当事者の識別子との関数と結合されることもあるからである。

【００１３】本発明の一実施例によれば、当事者は、少なくともいくつかのパラメータの関数を計算し、計算した値を他方当事者に送信し、受信された値を各当事者が自己の計算値に照らしてチェックすることによって、互いを認証することが可能である。この計算に用いられるパラメータは、少なくとも１つの当事者識別子、Diffie-Hellman値（g^xまたはg^y）、共有秘密、および共有パスワードとすることが可能である。これらの値のうちの少なくとも１つの関数を計算することにより、当事者は、他方当事者が共有パスワードを所有していることを認証することが可能である。

【００１４】本発明のもう１つの実施例によれば、当事者は、明示的に互いを認証するのではなく、代わりに、当事者は、それぞれ共有秘密鍵を生成し、生成した共有秘密鍵を通信のためのセッション鍵として使用することによって、暗黙的に互いを認証する。いずれかの当事者が正しいパスワードを所有していない場合、その当事者は正しい秘密セッション鍵を生成することができず、当事者間の通信は不可能となる。この実施例によれば、両当事者が、群演算を用いて自己のDiffie-Hellman値を少なくともパスワードの関数と結合し、結果として得られる値を他方当事者へ送信するという上記の技術を使用する。他方当事者から値を受信すると、各当事者は、群逆演算を用いて他方当事者のDiffie-Hellman値を抽出し、共有秘密鍵を計算する。

【００１５】通信プロトコルに対する二当事者は、ほとんどの場合、クライアントコンピュータとサーバコンピュータである。上記の実施例では、クライアントおよびサーバはいずれも共有パスワードを記憶する。本発明の他の実施例では、サーバにおけるセキュリティ劣化に対する保護を行うため、サーバは、パスワードを所有せず、代わりに、いわゆるパスワードベリファイアを与えられ、記憶する。パスワードベリファイアは、以下で詳細に説明するように、パスワードのある関数である。パスワード自体は、パスワードベリファイアの知識から決定することはできない。本発明のこれらの実施例によるプロトコルは、パスワードベリファイアが一般に実際のパスワードの代わりに用いられることを除いては、上記の実施例と類似している。しかし、サーバが実際のパスワードを知らないため、他方当事者が実際に正しいパスワードベリファイアまたは実際のパスワードを所有していることを各当事者が安全に認証するためには、異なる技術が二当事者によって使用されなければならない。一実施例では、当事者は、エルガマル暗号技術に基づく暗号化を用いて互いを認証する。

【００１６】

【発明の実施の形態】暗号方式は、二当事者間に安全な通信を提供するための周知の技術である。本発明のさまざまな実施例について説明する前に、いくつかの背景的知識および基本的用語について説明する。

【００１７】略式には、集合Ｓから集合Ｔへの関数ｆについて、Ｓに属するすべてのｘに対してはｆ（ｘ）は計算が容易であるが、Ｔに属するほとんどのｙに対しては、ｆ（ｘ）＝ｙとなるようなＳの元ｙを見つけるのが計算量的に実行不能である場合、ｆは一方向性関数であるという。一方向性関数の一例は、法指数演算である。ｐを大きい素数とし、ｇを、ｐを法とする乗法群（すなわち、１，．．．，ｐ－１の範囲の数）の生成元とする。この場合、$f(x) = g^x \bmod p$は一般に、一方向性関数であると仮定される。その逆関数（離散対数関数と呼ばれる）は、計算が困難である。離散対数関数の計算が困難な他の群もある。例えば、ある種の楕円曲線群である。

【００１８】ｋおよびｌをセキュリティパラメータとする。ただし、ｋは、主セキュリティパラメータであり、ハッシュ関数および秘密鍵に対する一般的なセキュリティパラメータとみなされ、ｌ＞ｋは、離散対数に基づく公開鍵に対するセキュリティパラメータとみなされる。$\{0,1\}^*$を、有限二進数列の集合とし、$\{0,1\}^n$を、長さｎの二進数列の集合とする。実数値関数$\varepsilon(n)$は、任意のｃ＞０について、すべてのｎ＞n_cに対して$\varepsilon(n) < 1/n_c$となるような$n_c > 0$がある場合、無視しうるという。少なくともサイズｋのｑと、サイズｌのｐを、ある値ｒに対してｐ＝ｒｑ＋１がｑと互いに素であるような素数とする。ｇを、Z_p^*の、サイズｑの部分群の生成元とする。この部分群を$G_{p,q}$と呼ぶことにする。

【００１９】Diffie-Hellman鍵交換と呼ばれる鍵交換プロトコル（W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Informati

on Theory, vol.22, no.6, 644-654, 1976、に記載）は、法指数関数に基づいている。具体的には、二当事者ＡとＢが、図１に記載するプロトコルに従って、秘密鍵について合意する。ステップ１０２で、Ａは、群Z_qからランダムにｘを選ぶ。ただし、Z_q＝｛０，１，...，ｑ－１｝（あるいは単に、ｑを法とする整数）である。ステップ１０４で、Ａは、Ｘ＝g^x　ｍｏｄ　ｐを計算する。ステップ１０６で、Ａは、ＸをＢへ送信する。ステップ１０８で、Ｂは、Z_qからランダムにｙを選ぶ。Ｂは、ステップ１１０で、Ｙ＝g^y　ｍｏｄ　ｐを計算し、ステップ１１２で、ＹをＡへ送信する。この時点で、共有秘密g^{xy}（すなわち、秘密鍵）が、ＡとＢの両者により計算可能となる。（なお、以下、ｍｏｄ　ｐで考えていることが明らかである場合には、記法を簡単にするため、ｍｏｄ　ｐの記載を省略することがある。）Ｘ＝g^xがステップ１０６でＡからＢへ送信されたため、ステップ１１６で、Ｂは、X^yを計算することにより、共有秘密g^{xy}を計算することができる。同様に、Ｙ＝g^yがステップ１１２でＢからＡへ送信されたため、ステップ１１４で、Ａは、Y^xを計算することにより、共有秘密g^{xy}を計算することができる。こうして、共有秘密Ｓは、安全な通信のためのセッション鍵として、ＡとＢにより用いられることが可能となる。

【００２０】Diffie-Hellman鍵交換は、ある種の楕円曲線群のような、離散対数関数の計算が困難な他の群の上でも実行可能である。群は当業者に周知であり、I. N. Herstein, "Topics in Algebra", 2nd edition, John Wiley and Sons, New York, 1975、には次のように記載されている。空でない元の集合Ｇにおいて、積と呼ばれ・で表される次のような二項演算が定義される場合、Ｇは群をなすという。

１．ａ，ｂ∈Ｇならば、ａ・ｂ∈Ｇ（閉じている）。

２．ａ，ｂ，ｃ∈Ｇならば、ａ・（ｂ・ｃ）＝（ａ・ｂ）・ｃ（結合律）

３．すべてのａ∈Ｇに対して、ａ・ｅ＝ｅ・ａ＝ａとなるような元ｅが存在する（Ｇにおける単位元の存在）。

４．任意のａ∈Ｇに対して、ａ・a^{-1}＝a^{-1}・ａ＝ｅとなるような元a^{-1}が存在する（Ｇにおける逆元の存在）。

【００２１】こうして、より一般的に、Diffie-Hellman鍵交換は、特定の群において作用する。ただし、秘密鍵ｘおよびｙは、群の元の指数である。そこで、ｇ∈Ｇを群生成元とする群Ｇ＝｛ｇ，ｇ・ｇ，ｇ・ｇ・ｇ，ｇ・ｇ・ｇ・ｇ，...｝を考える。ここで、・は群演算である。例として、Ｇの群演算が乗算である場合、Ｇ＝｛g^1，g^2，g^3，g^4，...｝である。Ｇの群演算が加算である場合、Ｇ＝｛１ｇ，２ｇ，３ｇ，４ｇ，...｝である。本発明は、さまざまな群を用いて実施可能であるため、以下の説明および特許請求の範囲において、記法g^xは、群生成元ｇに対して群演算をｘ回作用させることを意味する。さらに、任意の群について、ここで／で表す群逆演算も存在する。以下の説明および特許請求の範囲において、群逆演算は次のように定義される。すなわち、ｘおよびｙに対する群逆演算、すなわち、ｘ／ｙは、ｘ・y^{-1}として定義される。

【００２２】本発明の第１実施例による相互認証および鍵交換プロトコルを図２に示す。図の左側に示すステップは第１当事者Ａによって実行され、図の右側に示すステップは第２当事者Ｂによって実行される。通常、Ａはクライアントマシンとなり、Ｂはサーバマシンとなる。しかし、これは必ずしも必要ではなく、ＡおよびＢは、代表的な場合を示す例としてそれぞれクライアントおよびサーバとラベルされているだけである。理解されるように、本発明は、ＡおよびＢがクライアントおよびサーバである場合に限定されるものではなく、任意の二当事者ＡおよびＢに適用可能である。矢印は、当事者間の通信を表す。このプロトコルによれば、サーバは自分自身をクライアントに認証してもらい、クライアントは自分自身をサーバに認証してもらう。両者の認証が完了した後、それぞれが、後の安全な通信に使用可能な秘密セッション鍵を生成することになる。

【００２３】プロトコルを開始する前に、クライアントおよびサーバは、クライアントがサーバとの認証に用いるパスワードπを所有すると仮定する。

【００２４】なお、以下のプロトコルは、サーバおよびクライアントの両方を認証する。したがって、サーバおよびクライアントのいずれも真正でないと仮定され、このため、サーバまたはクライアントのいずれかが敵である可能性がある。クライアントは、自分自身を認証してもらってサーバにアクセスしようとする敵である可能性がある。サーバは、疑いを持たないクライアントから要注意情報を得ようとして別の真正なサーバのふりをしようとする敵である可能性がある。

【００２５】当業者には容易に明らかになるように、サーバおよびクライアントは、コンピュータプログラムコードの制御下で動作するプログラムされたコンピュータとして実装可能である。コンピュータプログラムコードは、コンピュータ可読媒体（例えば、メモリ）に記憶され、コードは、コンピュータのプロセッサにより実行される。本発明の明細書の記載に基づいて、当業者であれば、ここで説明するプロトコルを実装するために適当なコンピュータプログラムコードを容易に作成することが可能である。クライアントとサーバは、データネットワークを通じて互いに通信する。このようにネットワーク接続された、プログラムされたコンピュータは当業者に周知であるため、ここでは詳細には説明しない。

【００２６】図２に戻り、ステップ２０２で、クライアントは、Z_qから指数ｘとしてランダムな値を選ぶ。次に、ステップ２０４で、クライアントは、ｍ＝g^x・

（H_1（A，B，π））^{r} ｍｏｄ　ｐとして、パラメータｍを計算する。ただし、Ａはクライアントの一意的識別子であり、Ｂはサーバの一意的識別子であり、πは、この特定のサーバに対するクライアントのパスワードであり、H_1はランダムハッシュ関数であり、・は群演算を表す。H_1（Ａ，Ｂ，π）は、結果が$G_{[p,q]}$内にあることを保証するために、ｒ乗される。略式には、集合Ｓから集合Ｔへの関数Ｈについて、Ｈの出力がランダムに見える場合、すなわち、少なくとも、Ｓに属する入力ｘで関数が計算されるまでは予測不能である場合、ランダムハッシュ関数と呼ばれる。H_1は、Z_p^*においてランダムに見えるものを出力しなければならないため、｜ｐ｜＋ｓｅｃビット（ただし、｜ｐ｜はｐのビット数であり、ｓｅｃはセキュリティパラメータである）を出力すべきである。セキュリティパラメータは、例えば、１６０とすることが可能である。一般にこのようにふるまう既知の関数には次のものがある。

・ＳＨＡ－１（FIPS 180-1, "Secure Hash Standard", Federal InformationProcessing Standards Publication 180-1, 1995、に記載）

・ＲＩＰＥＭＤ－１６０（H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160: a strengthened version of RIPEMD", Fast Software Encryption, 3rdIntl. Workshop, 71-82, 1996、に記載）

【００２７】各クライアント・サーバ対ごとに一意的であることを保証するために、パスワードのみではなく、組（タプル）（Ａ，Ｂ，π）を用いる。発見的セキュリティに要求されると思われるのはパスワードのみであるが、以下で詳細に説明するように、セキュリティの形式的証明には、クライアントおよびサーバの名前が必要であると思われる。したがって、本発明の１つの特徴によれば、少なくともパスワードの関数と、Diffie-Hellman値g^xとに対して群演算を実行することにより、少なくともパスワードの関数をDiffie-Hellman値g^xと結合する。これは、パスワードの知識を有する者によってのみDiffie-Hellman値g^xがパラメータｍから抽出可能であることを保証するので、プロトコルの重要なステップである。このDiffie-Hellman値g^xの抽出については、ステップ２１４に関して以下で詳細に説明する。ステップ２０６で、クライアントは、パラメータｍをサーバへ送信する。

【００２８】パラメータｍを受信すると、ステップ２０８で、サーバは、パラメータ値が０ｍｏｄ　ｐでないことを保証するために、パラメータ値をテストする。この値が０　ｍｏｄ　ｐである場合、サーバはプロトコルを終了する。０はZ_p^*に属さないからである。この値が０　ｍｏｄ　ｐでない場合、ステップ２１０で、サーバは、Z_qから指数ｙとしてランダムな値を選ぶ。ステップ２１２で、サーバは、パラメータμにDiffie-Hellman値g^yを代入する。次に、ステップ２１４で、サーバは、受信したパラメータｍを用いて、次式のように、Diffie-Hellman共有秘密g^{xy}（このプロトコルではσで表す）を計算する。

$\sigma = (m / (H_1(A, B, \pi))^r)^y \mod p$

次に、このステップについて、さらに詳細に説明する（記法を簡単にするためｍｏｄ　ｐの記載を省略する）。まず、想起されるべき点であるが、上記のように、任意の群演算に対して、ｘおよびｙに対する群逆演算、すなわち、ｘ／ｙがｘ・y^{-1}として定義されるような群逆演算がある。そこで、当業者には認識されるように、ステップ２１４におけるｍ／（H_1（Ａ，Ｂ，π））^{r}の計算は、ｍと、少なくともパスワードの関数とに対する群逆演算を実行している。ステップ２０４からのｍの値を代入すると、$g^x \cdot (H_1(A, B, \pi))^r / (H_1(A, B, \pi))^r = g^x$を得る。したがって、サーバは、正しいパスワードπを所有している場合、受信したパラメータｍの値からDiffie-Hellman値g^xを抽出することができる。こうして、ステップ２１４における計算の結果、サーバは、Diffie-Hellman共有秘密g^{xy}を生成する。

【００２９】次に、ステップ２１６で、サーバは、ｋ＝H_{2a}（Ａ，Ｂ，ｍ，μ，σ，π）を計算する。ただし、H_{2a}は、ｓｅｃビット（ただしｓｅｃはセキュリティパラメータ）を出力しなければならないもう１つのランダムハッシュ関数である。パラメータｋは、以下で説明するように、サーバが正しいパスワードを所有していることを認証するために、クライアントＡにより使用されることになる。ステップ２１８で、サーバは、パラメータμおよびｋをクライアントへ送信する。

【００３０】パラメータμおよびｋを受信すると、クライアントは、ステップ２２０で、$\sigma = \mu^x \mod p$を計算する。$\mu = g^y$であるため、$\mu^x = g^{xy}$はDiffie-Hellman共有秘密である。ステップ２２２で、クライアントは、自己のπの知識を用いてH_{2a}（Ａ，Ｂ，ｍ，μ，σ，π）を計算し、その結果が、ステップ２１８でサーバから受信したパラメータｋに等しいかどうかをテストする。これらが等しい場合、クライアントはサーバを認証したことになる。これらが等しくない場合、サーバは認証されなかったので、クライアントはプロトコルを終了する。ステップ２２４で、クライアントはｋ′＝H_{2b}（Ａ，Ｂ，ｍ，μ，σ，π）を計算する。これは、以下で説明するように、サーバがクライアントを認証するために用いられることになる。ステップ２２６で、クライアントは、Ｋ＝H_3（Ａ，Ｂ，ｍ，μ，σ，π）としてセッション鍵Ｋを生成する。ステップ２２８で、クライアントは、ｋ′をサーバへ送信する。今度も、H_{2b}およびH_3は、ｓｅｃビット（ただしｓｅｃはセキュリティパラメータ）を出力しなければならないランダムハッシュ関数である。

【００３１】ステップ２３０で、サーバは、自己のπの

知識を用いてH_{2b}（A，B，m，μ，σ，π）を計算し、その結果が、ステップ２２８でサーバから受信したパラメータｋ′に等しいかどうかをテストする。これらが等しい場合、サーバはクライアントを認証したことになる。これらが等しくない場合、クライアントは認証されなかったので、サーバはプロトコルを終了する。ステップ２３２で、サーバは、Ｋ＝H_3（A，B，m，μ，σ，π）としてセッション鍵Ｋを生成する。

【００３２】この時点で、クライアントとサーバの両者は互いを認証したことになり、クライアントとサーバはいずれも同一の安全なセッション鍵Ｋを生成している。これは、クライアントとサーバの間の後の安全な通信に用いることが可能である。

【００３３】次に、本発明の第２実施例について図３を参照して説明する。この実施例では、当事者間の認証は暗黙的である。これは、一方当事者が他方当事者を明示的に認証するステップがないことを意味する。むしろ、各当事者はセッション鍵を生成しようとするが、いずれかの当事者が正しいパスワードを有しない場合、その当事者は正しいセッション鍵を生成することができず、当事者間の通信が不可能となる。図３を参照すると、ステップ３０２～３１０は、図２のステップ２０２～２１０に対応する。ステップ３１２で、サーバは、$\mu = g^{y} \cdot (H_2(A, B, \pi))^{r} \bmod p$としてパラメータ$\mu$を計算する。これは、図２に関して前述したステップ２０４と同様である。ステップ２０４に関して前述したように、実際には、ステップ３１２は、少なくともパスワードの関数と、Diffie-Hellman値g^{y}とに対して群演算を実行することにより、少なくともパスワードの関数をDiffie-Hellman値g^{y}と結合することになる。

【００３４】次に、ステップ３１４で、サーバは、ステップ２１４に関して前述したのと同様に、Diffie-Hellman値g^{x}を抽出し、Diffie-Hellman共有秘密g^{xy}を計算する。ステップ３１６で、サーバは、Ｋ＝H_3（A，B，m，μ，σ，π）としてセッション鍵Ｋを計算する。ステップ３１８で、サーバは、ステップ３１２で生成したパラメータμをクライアントへ送信する。

【００３５】パラメータμを受信すると、ステップ３２０で、クライアントは、パラメータμの値が$0 \bmod p$でないことを保証するために、このパラメータ値をテストする。この値が$0 \bmod p$である場合、クライアントはプロトコルを終了する。この値が$0 \bmod p$でない場合、ステップ３２２で、クライアントは、μの値を用いて、前述のように、Diffie-Hellman値g^{y}を抽出し、Diffie-Hellman共有秘密g^{xy}を計算する。ステップ３２４で、クライアントは、Ｋ＝H_3（A，B，m，μ，σ，π）としてセッション鍵Ｋを計算する。

【００３６】この時点で、クライアントおよびサーバはいずれも、パスワードの知識に基づいて、セッション鍵Ｋを生成したことになる。クライアントおよびサーバが正しいパスワードを所有していたとすれば、両者は同じセッション鍵Ｋを生成していることになり、これを用いて安全な通信が可能である。しかし、当事者の一方が正しいパスワードを所有していなかった場合、その当事者は、正しいセッション鍵を生成しておらず、当事者間の通信が不可能となる。

【００３７】図２および図３に関して説明したプロトコルは、サーバがパスワードπを所有し記憶していることを仮定している。このようなプロトコルの１つの潜在的な問題点は、サーバ記憶領域のセキュリティ劣化により、敵がクライアントのパスワードを取得する可能性があることである。このような事態に対する保護を行うため、次に説明する本発明のもう１つの実施例では、サーバはパスワードπを所有せず、代わりに、いわゆるパスワードベリファイアＶを記憶する。ここで、パスワードベリファイアとは、パスワードの関数として計算可能な値であるが、パスワードをパスワードベリファイアの知識から決定することはできない。ここで説明する実施例では、パスワードベリファイアＶは、パスワードπのある関数である。ただし、特定のクライアントＡおよびサーバＢに対して、Ｖ＝g^{v}、ｖ＝H_0（A，B，π）であり、H_0は、｜ｑ｜＋ｓｅｃビットを出力しなければならないランダムハッシュ関数である。クライアントは、πを知っているため、ｖ＝H_0（A，B，π）を計算することができ、さらに、ｖからベリファイアＶ＝g^{v}を計算することができる。サーバはＶしか知らないため、ｖ（これは、Ｖの離散対数である）を計算することはできない。本発明のこの実施例によるプロトコルを図４に示す。まず、ステップ４０１で、クライアントは、上記のようにパスワードベリファイアＶを生成する。ステップ４０２～４２２は、ステップ２０２～２２２に関して前述したように実行される。ただし、ステップ４０２～４２２では、パスワードπをパスワードベリファイアＶで置き換える。この時点で、ステップ４２２でのテストが真である場合、クライアントは、サーバが正しいパスワードベリファイアＶを知っていることを認証したことになり、次に、クライアントは、自分が正しいパスワードを知っていることを証明することによって、自分自身をサーバに認証してもらわなければならない。

【００３８】ステップ４２４で、クライアントは、Z_qから指数Ｃとしてランダムな値を選ぶ。クライアントは、ステップ４２６で、ａ＝g^{C}を計算し、ステップ４２８で、ｅ＝Ｈ（Ａ，Ｂ，ｍ，μ，σ，ａ，Ｖ）を計算する。次に、ステップ４３０で、クライアントは、Ｓ＝Ｃ－ｅｖを計算する。ステップ４３２で、クライアントは、Ｓおよびａをサーバへ送信する。Ｓおよびａを受信すると、ステップ４３４で、サーバは、自己のＶの知識を用いて、ｅ＝Ｈ（Ａ，Ｂ，ｍ，μ，σ，ａ，Ｖ）を計算する。ステップ４３６で、サーバは、$g^{S}V^{e}$を計算

する。ステップ４３６でサーバにより計算された値が、ステップ４３２でクライアントから受信したａの値と一致する場合、サーバは、クライアントを真正なものとして受け入れる。最後に、クライアントおよびサーバは、それぞれステップ４３８および４４０で、Ｋ＝Ｈ$_3$（Ａ，Ｂ，ｍ，μ，σ，ａ，Ｖ）としてセッション鍵を計算する。

【００３９】直観的には、ステップ４３６でのテストは、クライアントを次のようにして認証することになる。ステップ４３６でのａ＝ｇ^{S}Ｖ^{e}というサーバの計算を参照すると、Ｖ＝ｇ^{v}であるため、ステップ４３６での計算は、ａ＝ｇ^S（ｇ^v）^{e}＝ｇ^{S}ｇ^{ev}＝ｇ$^{S+ev}$＝ｇ^{C}となる。ステップ４３０から、Ｃ＝Ｓ＋ｅｖであることがわかるからである。したがって、サーバによるａの計算結果が、ステップ４３２でクライアントから受信したａと一致した場合、サーバは、クライアントがｖの知識を有していることを知る。ｖは、クライアントがパスワードπを知っている場合にのみ計算することができたものである。

【００４０】本発明のもう１つの実施例を図５に示す。これも、サーバがパスワードπを所有せず、代わりに、パスワードベリファイアＶを記憶するようなプロトコルを示す。ステップ５０１～５１４は、図４に関して前述したステップ４０１～４１４と同様である。ステップ５１６で、サーバは、Ｚ$_q$からランダムなｃを選び、ａを次のように計算する。

【数１】

$$c \in_R \{0,1\}^k, a = g^{H_0'(A,B,c)} \bmod p$$

ただし、Ｈ$_0$′は、｜ｑ｜＋ｓｅｃビットを出力しなければならないあるランダムハッシュ関数である。ステップ５１８で、サーバは、次式を計算する。

【数２】

$$k = c \oplus H_{2a}(A, B, m, \mu, a, \sigma, V^{H_0'(A,B,c)} \bmod p, V)$$

ただし、

【数３】

$$\oplus$$

は排他的論理和（ＸＯＲ）を表す。ステップ５２０で、サーバは、μ，ａ，ｋをクライアントへ送信する。μ，ａ，ｋを受信すると、ステップ５２２で、クライアントは、σ＝μ^x　ｍｏｄ　ｐを計算する。この時点で、クライアントおよびサーバはいずれも、共有秘密σを所有する。ステップ５２４で、クライアントは、次式を計算する。

【数４】

$$c = k \oplus H_{2a}(A, B, m, \mu, a, \sigma, a^v, V)$$

ステップ５２６で、クライアントは、

【数５】

$$g^{H_0'(A,B,c)} \bmod p$$

を計算し、計算した値が、ステップ５２０でサーバから受信したａの値と同一であるかどうかをテストする。同一でない場合、サーバはクライアントに正しく認証されておらず、クライアントはプロトコルを終了する。計算されたａがステップ５２０でサーバから受信したａの値と同一である場合、クライアントは、サーバが真正なものであると判断し、プロトコルのステップ５２８に進み、ｋ′＝Ｈ$_{2b}$（Ａ，Ｂ，ｍ，μ，σ，ａ，ｋ，ｃ，Ｖ）を計算する。ステップ５３０で、クライアントは、セッション鍵Ｋ＝Ｈ$_3$（Ａ，Ｂ，ｍ，μ，σ，ｃ，Ｖ）を計算する。ステップ５３２で、クライアントは、ｋ′をサーバへ送信する。ステップ５３４で、サーバは、Ｈ$_{2b}$（Ａ，Ｂ，ｍ，μ，σ，ａ，ｋ，ｃ，Ｖ）を計算し、計算した値が、ステップ５３２でクライアントから受信したｋ′の値と同一であるかどうかを判断する。同一でない場合、クライアントはサーバに認証されておらず、サーバはプロトコルを終了する。計算されたｋ′が、ステップ５３２でクライアントから受信したｋ′の値と同一である場合、サーバは、クライアントが真正なものであると判断し、プロトコルのステップ５３６に進み、セッション鍵Ｋ＝Ｈ$_3$（Ａ，Ｂ，ｍ，μ，σ，ｃ，Ｖ）を計算する。

【００４１】直観的には、クライアントとサーバの認証はエルガマル暗号に基づいている。ここではエルガマル暗号については簡単に説明するにとどめるが、さらに詳細には、T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Basedon Discrete Logarithms", IEEE Transactions on Information Theory, IT-31,4, pp.469-472, 1985、に記載されている。一般に、メッセージＭは、エルガマル暗号によりＥ（Ｍ）＝（ｇ^r，ｙ^{r}Ｍ）と暗号化される。ただし、ｒはランダムな値であり、ｙは公開鍵であり、ｘを秘密鍵として、ｙ＝ｇ^{x}である。暗号（Ａ，Ｂ）は、Ｄ（Ａ，Ｂ）＝Ｂ／Ａ^{x}として復号される。値を代入すると、Ｂ／Ａ^x＝ｙ^{r}Ｍ／ｇ^{rx}＝ｇ^{xr}Ｍ／ｇ^{rx}＝Ｍとなる。このように、エルガマル方式を用いて暗号化されたメッセージを復号するためには、秘密鍵が必要である。エルガマル暗号の第２版では、メッセージＭは、次式のように暗号化される。

【数６】

$$E(M) = (g^r, H(y^r) \oplus M)$$

ただし、ｒはランダムな値であり、ｙは公開鍵であり、

ｘを秘密鍵として、ｙ＝ｇ^{x}である。暗号（Ａ，Ｂ）は、この第２方式によれば、次式のように復号される。

【数７】

$$D(A,B) = H(A^x) \oplus B$$

値を代入すると、次式を得る。

【数８】

$$H(A^x) \oplus H(y^r) \oplus M = H(y^r) \oplus H(y^r) \oplus M = M$$

こうして、この場合も、エルガマル第２方式を用いて暗号化されたメッセージを復号するためには、秘密鍵が必要である。

【００４２】ここで、「自己証明エルガマル暗号」を定義する。この暗号方式では、エルガマルの第２版を用いるが、ｒは、単なる純粋にランダムな値の代わりに、ｒ＝Ｈ′（Ｍ）、すなわち、復号される値のランダムハッシュとする。この場合、当事者が、暗号（Ａ，Ｂ）を受信し、それを復号してＭを得ると、その当事者は、次式が成り立つかどうかをテストすることができる。

【数９】

$$A = g^{H'(M)}$$

注意すべき点であるが、任意の（Ａ，Ｂ）は何かの暗号を定義するが、自己証明は、何がまさに暗号化されたかを、したがって、どの暗号化鍵が用いられているかを、暗号化をした者が知っていることを証明するものである。

【００４３】ここで、図５に示したプロトコルを参照すると、当業者には認識されるように、ステップ５１６および５１８は、自己証明エルガマル暗号化を実行している。暗号化されているメッセージはランダム値ｃであり、公開鍵はパスワードベリファイアＶであり、秘密鍵はｖである。暗号値はパラメータｋに代入されている。ステップ５２４で、クライアントは、サーバから受信した、暗号化されたｋを復号してｃを抽出した後、ステップ５２６のテストを実行する。このテストが真である場合、サーバが正しいパスワードベリファイアＶを所有していることが証明される。ステップ５２８で、クライアントは、ｃの計算値を用いてｋ′を生成し、ステップ５３２でｋ′をサーバへ送信する。ステップ５３４でのサーバによるテストが真である場合、クライアントが正しいｖを所有していることが証明される。秘密鍵ｖがなければ、クライアントは復号を実行して、ｋを計算する際に用いられるハッシュ関数への入力であるｃを得ることができないからである。

【００４４】本発明のさらに別の実施例では、図２に示したプロトコルは、サーバがパスワードπを記憶せず、代わりに、値（Ｈ$_{1}$（Ａ，Ｂ，π））ｒを記憶するように修正することが可能である。これは、追加の計算を必要とせず、２つのサーバＢ$_{1}$およびＢ$_{2}$でπを使用する素朴なユーザは、Ｂ$_{1}$に欠陥が生じてもＢ$_{2}$では自明には脆弱にならないという利点がある。

【００４５】本発明の発明者は、本発明による相互認証および鍵交換プロトコルが安全であることを証明した。証明の直観的な概略は次の通りである。直観的には、次のことを証明しなければならない。

【００４６】（１）パスワードを共有しプロトコルに従う二当事者が互いを認証し長い共有秘密を得る。

【００４７】（２）Diffie-Hellmanプロトコルが安全であることを仮定すると、本発明のプロトコルは、信頼される当事者との「理想世界プロトコル」と同程度に安全である。「理想世界プロトコル」では、正当な二当事者は、互いにコネクションを設定し、信頼される当事者に、このコネクションを安全にするための長い共有秘密を生成してもらうことができるが、敵は、設定されたコネクションごとに１回（安全にされる前に）共有パスワードを信頼される当事者に問い合わせることが可能である。（直観的には、これは、敵が、パスワードのランダム推測を行い、自分自身を認証してもらおうとすることをモデル化している。）

【００４８】（１）の部分は、プロトコルを見れば明らかである。

【００４９】（２）の部分は、より難しい。パスワードを知らずに、理想世界の信頼される当事者を用いるだけで、実プロトコルをシミュレートし、このシミュレーションを攻撃する敵は、実世界の実プロトコルを攻撃する敵とは区別不能であるようにすることができることを示す。（これは、"multi-party simulatability"（複数当事者シミュレート可能性）技法という、よく知られた暗号理論の証明技法であり、D. Beaver, "Secure Multiparty Protocols and Zero-Knowledge Proof Systems Tolerating a Faulty Minority", Journal of Cryptology, 4(2), pp.75-122, 1991、に記載されている。）

【００５０】技術的には、われわれのモデルは、すべてのハッシュ関数が完全にランダムであり、その関数が使用されるときにはいつも、シミュレータは入力を調べ、出力をセットすること（これらの出力がランダムにセットされる限り）を仮定している。

【００５１】われわれのシミュレータの一般的な考え方は、単に、互いに通信する正当な二当事者間の長い共有秘密を模造し、他のすべての会話で、パスワードに関する推測を（敵のハッシュ関数問合せを検査することにより）検出し、それらの推測を、信頼される当事者への「テストパスワード」問合せに変えようとするものである。これの難しい部分は、敵が、設定されたコネクションあたり複数回のパスワード推測を行うことができないことを示すことである。これを示すためには、敵にそれが可能であるとすると、Diffie-Hellman問題が解けてしまう（具体的には、（未知のｘおよびｙに対して）Ｘ＝ｇ^{x}、およびＹ＝ｇ^{y}となる値Ｘ，Ｙ，Ｚ∈Ｇをとり、Ｚ＝ｇ^{xy}であるかどうかを判定することができる）こ

とを示す。

【００５２】上記の詳細な説明は、単なる例示であって、限定的なものではないと理解されるべきである。ここで説明した実施例は、本発明の原理の単なる例示であり、当業者であれば、本発明の技術的範囲および技術思想から離れることなく、さまざまな変形例を考えることが可能である。例えば、上記のプロトコルにおいて、いくつかのパラメータがハッシュ関数を評価する際に用いられている。注意すべき点であるが、発見的セキュリティにはそのすべてのパラメータが必要とされるわけではなく、付加的パラメータは、プロトコルが形式的に安全であることを証明することを可能にするためのものである。例えば、図２のプロトコルのステップ２１６、２２２、２２４、２３０、２２６および２３２、図３のプロトコルのステップ３１６および３２４、図４のプロトコルのステップ４１６、４２２、４３８および４４０、ならびに図５のプロトコルのステップ５３０および５３６は、プロトコルが発見的に安全であるためには、ハッシュ関数においてパラメータσしか必要でないと考えられる。しかし、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。同様に、図４のプロトコルのステップ４２８および４３４は、プロトコルが発見的に安全であるためには、ハッシュ関数においてパラメータσおよびａしか必要でないと考えられる。しかし、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。同様に、図５のプロトコルのステップ５１８および５２４は、プロトコルが発見的に安全であるためには、パラメータσ、ａ、および

【数１０】

$$V^{H_0'(A,B,c)}$$

しか必要でないと考えられる。しかし、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。最後に、図５のプロトコルのステップ５２８および５３４は、プロトコルが発見的に安全であるためには、ハッシュ関数においてパラメータσおよびｃしか必要でないと考えられる。しかし、この場合も、上記の付加的パラメータの使用により、プロトコルは形式的に安全であることが証明される。

【００５３】

【発明の効果】以上述べたごとく、本発明によれば、安全であることが証明可能な、安全な、パスワードのみによる相互ネットワーク認証プロトコルが実現される。

【図面の簡単な説明】

【図１】従来のDiffie-Hellman鍵交換プロトコルを示す図である。

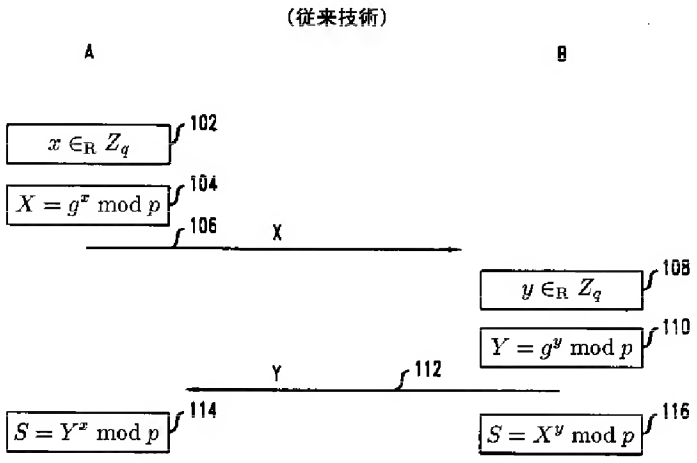
【図２】両当事者が共有パスワードを所有する、本発明の明示的認証実施例による通信プロトコルを示す図である。

【図３】両当事者が共有パスワードを所有する、本発明の暗黙的認証実施例による通信プロトコルを示す図である。

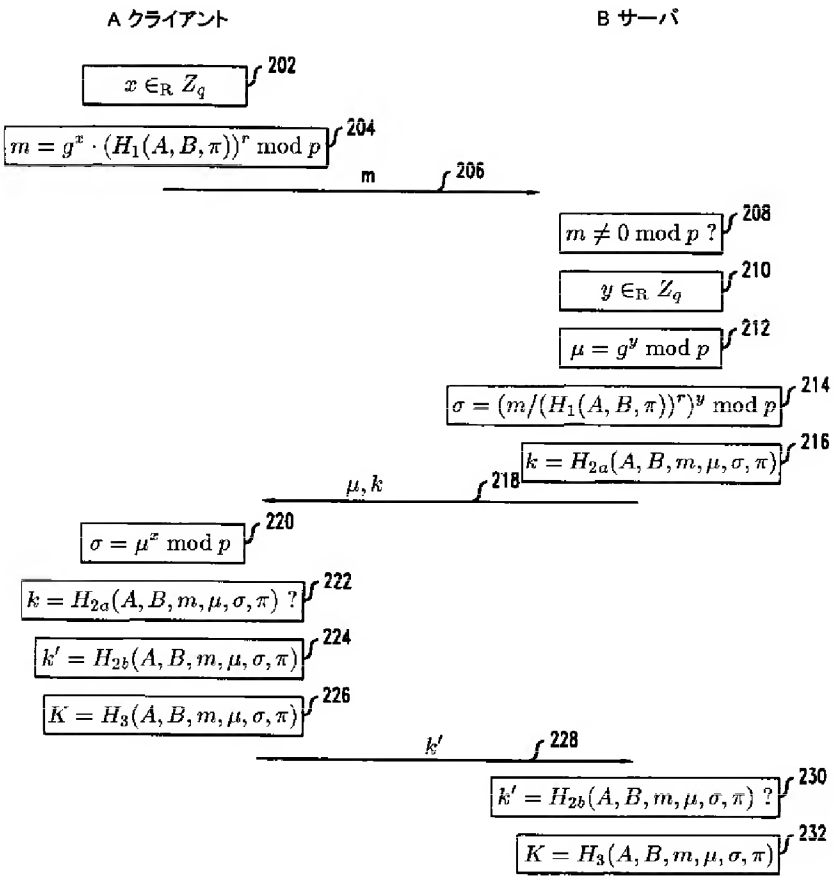
【図４】一方当事者がパスワードを所有し他方当事者がパスワードベリファイアを所有する、本発明の明示的認証実施例による通信プロトコルを示す図である。

【図５】一方当事者がパスワードを所有し他方当事者がパスワードベリファイアを所有する、本発明のもう１つの明示的認証実施例による通信プロトコルを示す図である。

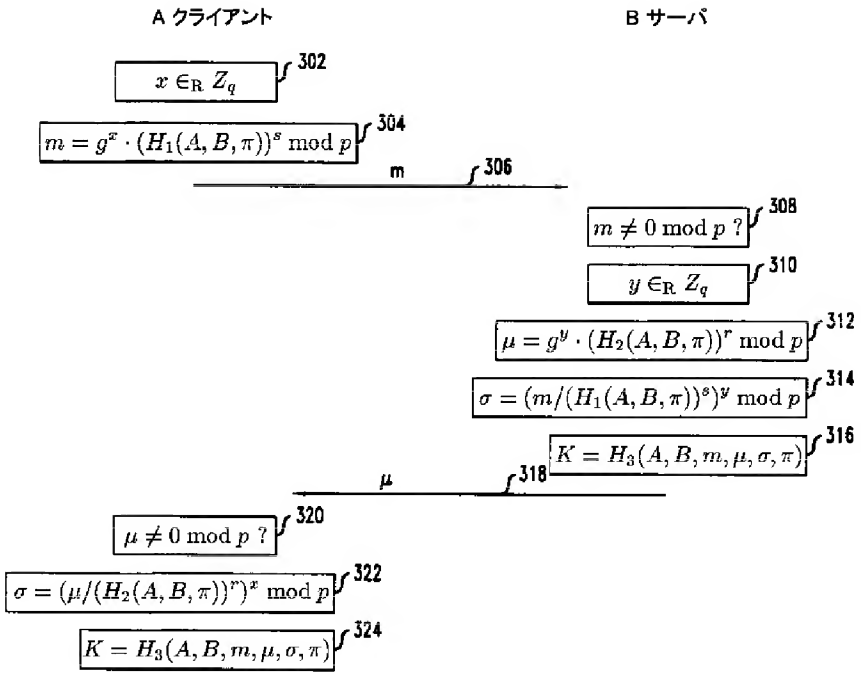
【図１】



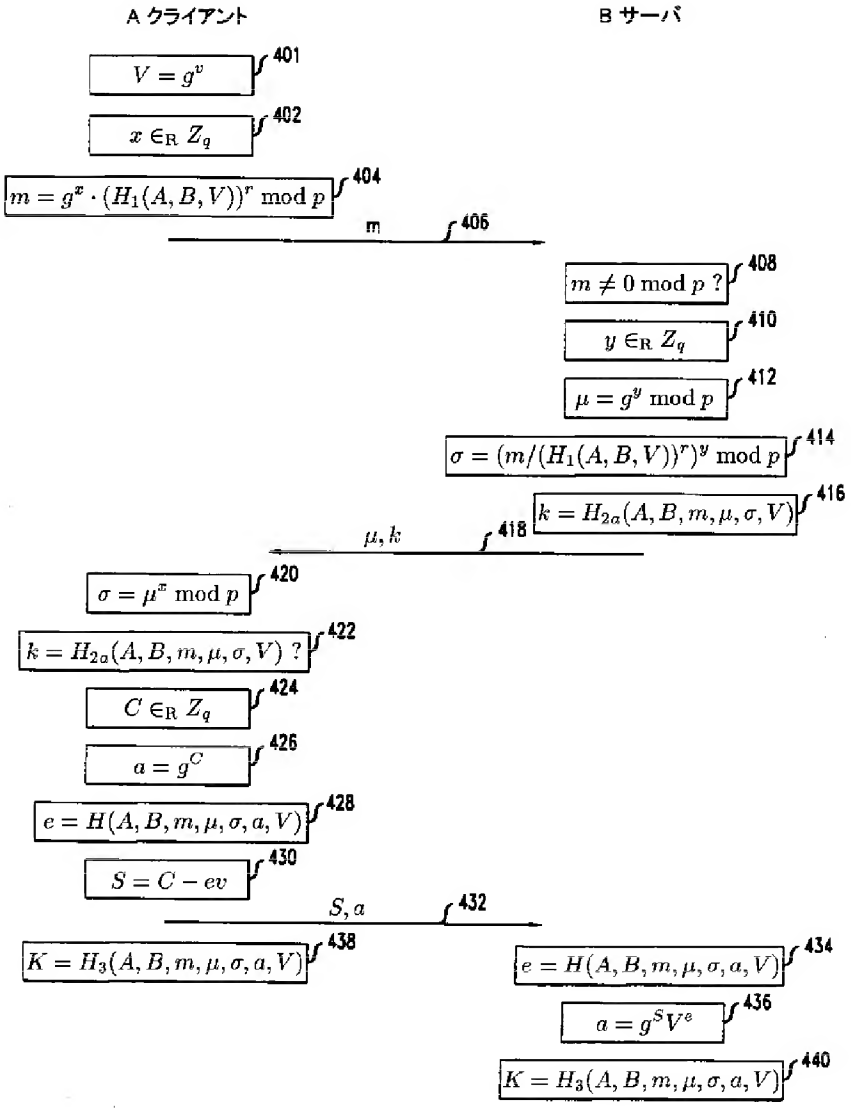
【図２】


A クライアント
B サーバ
$x \in_R Z_q$
202
$m = g^x \cdot (H_1(A,B,\pi))^r \bmod p$
204
m
206
$m \neq 0 \bmod p$?
208
$y \in_R Z_q$
210
$\mu = g^y \bmod p$
212
$\sigma = (m/(H_1(A,B,\pi))^r)^y \bmod p$
214
$k = H_{2a}(A,B,m,\mu,\sigma,\pi)$
216
μ, k
218
$\sigma = \mu^x \bmod p$
220
$k = H_{2a}(A,B,m,\mu,\sigma,\pi)$?
222
$k' = H_{2b}(A,B,m,\mu,\sigma,\pi)$
224
$K = H_3(A,B,m,\mu,\sigma,\pi)$
226
k'
228
$k' = H_{2b}(A,B,m,\mu,\sigma,\pi)$?
230
$K = H_3(A,B,m,\mu,\sigma,\pi)$
232

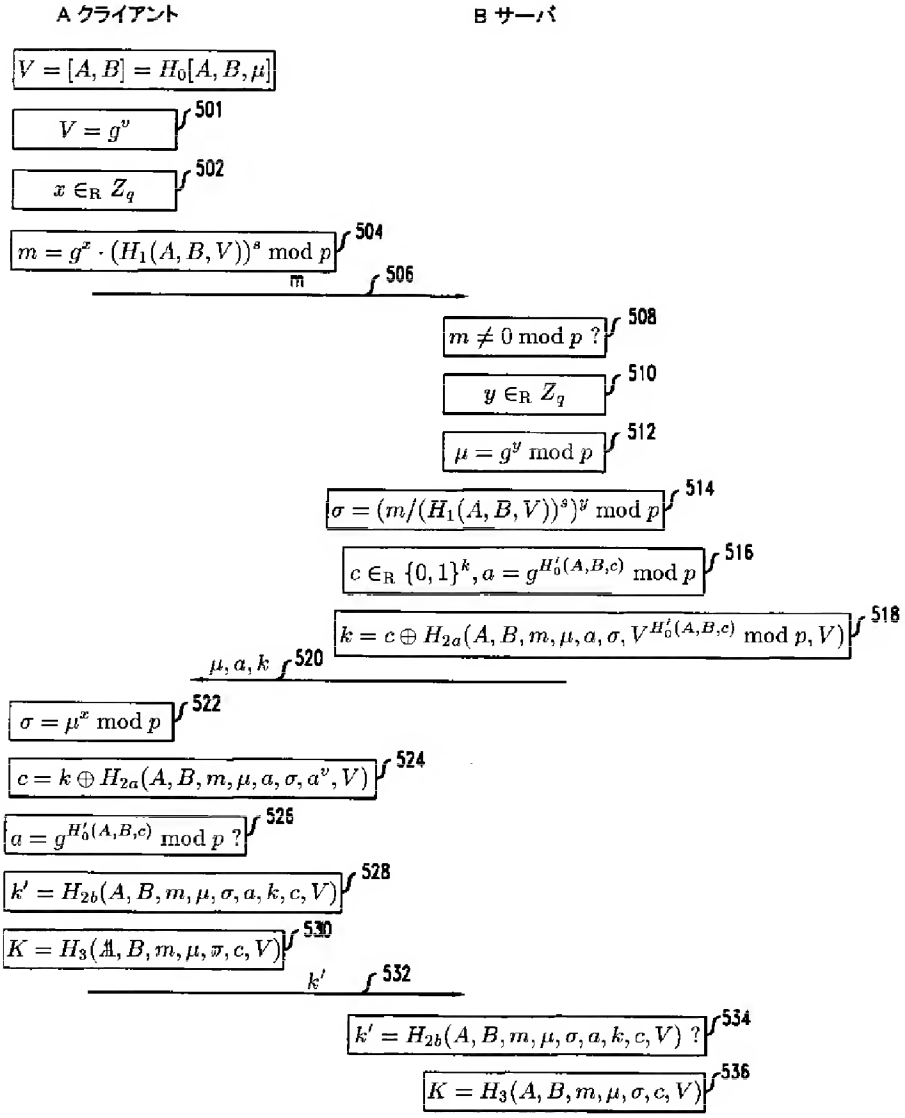
【図３】


A クライアント
B サーバ
$x \in_R Z_q$
302
$m = g^x \cdot (H_1(A,B,\pi))^s \bmod p$
304
m
306
$m \neq 0 \bmod p$?
308
$y \in_R Z_q$
310
$\mu = g^y \cdot (H_2(A,B,\pi))^r \bmod p$
312
$\sigma = (m/(H_1(A,B,\pi))^s)^y \bmod p$
314
$K = H_3(A,B,m,\mu,\sigma,\pi)$
316
μ
318
$\mu \neq 0 \bmod p$?
320
$\sigma = (\mu/(H_2(A,B,\pi))^r)^x \bmod p$
322
$K = H_3(A,B,m,\mu,\sigma,\pi)$
324

【図４】


A クライアント
B サーバ
$V = g^v$
401
$x \in_R Z_q$
402
$m = g^x \cdot (H_1(A,B,V))^r \bmod p$
404
m
406
$m \neq 0 \bmod p$?
408
$y \in_R Z_q$
410
$\mu = g^y \bmod p$
412
$\sigma = (m/(H_1(A,B,V))^r)^y \bmod p$
414
$k = H_{2a}(A,B,m,\mu,\sigma,V)$
416
μ, k
418
$\sigma = \mu^x \bmod p$
420
$k = H_{2a}(A,B,m,\mu,\sigma,V)$?
422
$C \in_R Z_q$
424
$a = g^C$
426
$e = H(A,B,m,\mu,\sigma,a,V)$
428
$S = C - ev$
430
S, a
432
$K = H_3(A,B,m,\mu,\sigma,a,V)$
438
$e = H(A,B,m,\mu,\sigma,a,V)$
434
$a = g^S V^e$
436
$K = H_3(A,B,m,\mu,\sigma,a,V)$
440

【図５】



フロントページの続き

(71)出願人　596077259
600 Mountain Avenue, Murray Hill, New Jersey 07974−0636U. S. A.

(72)発明者　エリック　グロッセ
アメリカ合衆国、07922、ニュージャージー州、バークリー　ハイツ、ノース　ロード　140

(72)発明者　ビクター　ウラジミール　ボイコ
アメリカ合衆国、10952、ニューヨーク州、マンジー、ルート306、244

(72)発明者　フィリップ　ディー、マッケンジー
アメリカ合衆国、07040　ニュージャージー州、メイプルウッド、カールトン　コート　11

(72)発明者　サーバー　パテル
アメリカ合衆国、07045　ニュージャージー州、モントビル、ミラー　レーン　34